Energy Considerations in Checkpointing and Fault Tolerance Protocols

M. Diouri, O. Gluck, L. Lefevre, F. Cappello

ENS Lyon/LIP Laboratory (ENS, INRIA, University of Lyon, CNRS)
JLPC (INRIA, NCSA, University of Illinois, ANL)

Boston (MA, USA), June 25^{th}, 2012

mehdi.diouri@ens-lyon.fr

2^{nd} Workshop Fault Tolerance for HPC at eXtreme-Scale

Context

An important growth of performance: a factor of 1000/10 years.

The most powerful supercomputer is IBM Sequoia (Top500): more than 1.5 billion cores and able to perform 16 PFlop/s.

A wide range of scientific applications:

IESP (USA), EESI (Europe): roadmaps to Exascale in 2018.

Context and Motivations

The issues addressed at the Exascale:

- Power and energy consumption
  - Most energy efficient: IBM BlueGene/Q [1]: 2 GFlops/W.
  - DARPA target: 20 MW for a 1 EFlop: 50 GFlops/W.
- Fault tolerance

  An exascale system = millions of cores.

  Faults many times per day.

  Fault tolerance is mandatory.

[1] Green 500: www.green500.org

Current Fault tolerance protocols

3 main categories of protocols:
uncoordinated, coordinated, hierarchical protocols.

Rely on checkpointing/restart:

- with message logging in uncoordinated protocols
- with process synchronization in coordinated protocols.

In hierarchical protocols: processes organized in clusters.

- process synchronization inside a same cluster.
- message logging between clusters.

Motivations

Both the issues of fault tolerance and power consumption are interrelated.

What are the power and energy consumption of current fault tolerance protocols ?

What is the best fault tolerance protocol in terms of power/energy consumption ?

Methodology

Both the issues of fault tolerance and power consumption are interrelated.

What are the power and energy consumption of current fault tolerance protocols ?
==> Experiments: benchmarks to study the energy behavior of the fault tolerance protocols.
==> 3 operations: Checkpointing, Message logging, Process coordination.

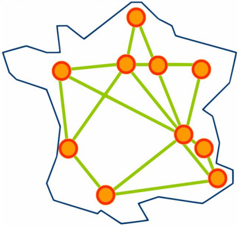
What is the best fault tolerance protocol in terms of power/energy consumption ?
==> Comparison of the energy consumption of fault tolerance operations during real applications (NAS).

Outline

Experimental infrastructure

Experiments on the Lyon site of Grid5000: a French scientific platform geographically distributed over 10 sites in France.

The Lyon site offers 64 available identical nodes Sun Fire V20z.



Each node gathers:

CPU	2 AMD Opteron 2.4 GHz, 1 core each
Memory	2 GB
Network	Gigabit Ethernet
HDD	SCSI, 73 GB

Experimental infrastructure

An energy-sensing infrastructure of external power meters from Omegawatt [2].

- instantaneous consumption in Watts;
- at each second for each monitored node;
- with a precision of 0.125 Watts.

We used only one core per node in all our experiments.
We ran each experiment 20 times.
We computed the mean value over the 20 values.

[2] http://www.omegawatt.fr/gb/index.php

Energy in fault tolerance protocols - Checkpointing

Checkpointing: storing a snapshot image of the current application state.

From the Berkeley Lab Checkpoint/Restart library (BLCR)[3]:

Available in MPICH2 [4].

A benchmark with one process and a 1GByte to checkpoint.

[3]BLCR: https://ftg.lbl.gov/projects/CheckpointRestart/
[4]MPICH2: http://www.mcs.anl.gov/research/projects/mpich2/

Energy in fault tolerance protocols - Checkpointing

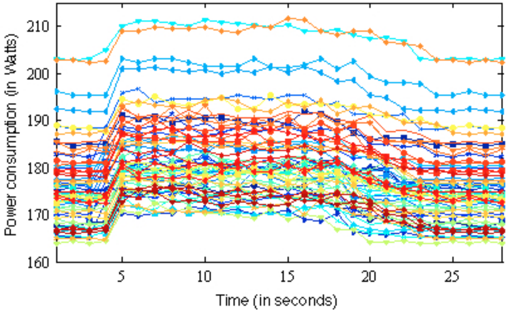


Figure: Power consumption due to a 1GByte checkpointing

Energy in fault tolerance protocols - Checkpointing

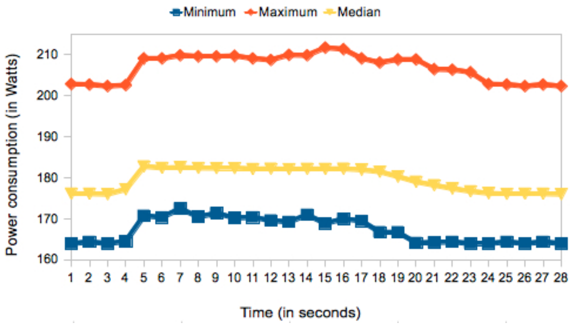


Figure: Power consumption due to a 1GByte checkpointing for the less/more/median consuming nodes

Energy in fault tolerance protocols - Checkpointing

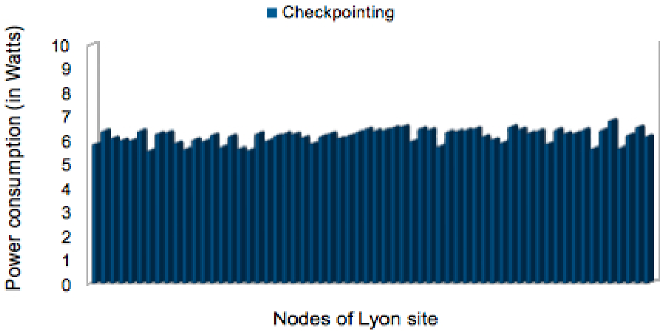


Figure: Extra power cost due to checkpointing

Energy in fault tolerance protocols - Checkpointing

Table: Energy ratio of checkpointing in Joules/GByte

Nodes	Checkpointing consumption
the less consuming	2520
the median consuming	2875
the more consuming	3570

The large difference between the less and the more consuming nodes is mainly due to:

- the difference in the idle power consumption for about 70 %
- the difference in the checkpointing duration for about 30 %.

Energy in fault tolerance protocols - Message logging

The sender process logs all the messages that are sent to other processes.

The logging function used each time a process sends a message.

We log 100,000 messages of 100 KBytes to get a total volume of 10 GBytes.

We ran the same benchmark for the 64 nodes.

Energy in fault tolerance protocols - RAM logging

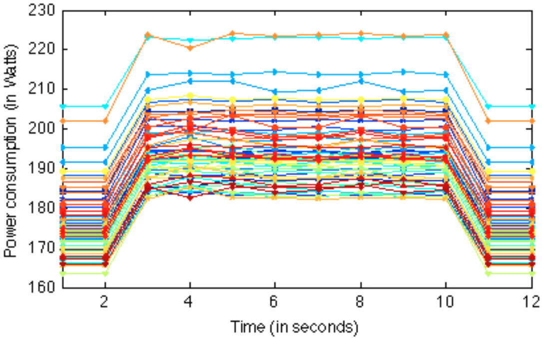


Figure: Power consumption of 10 GBytes of RAM Logging

Energy in fault tolerance protocols - HDD logging

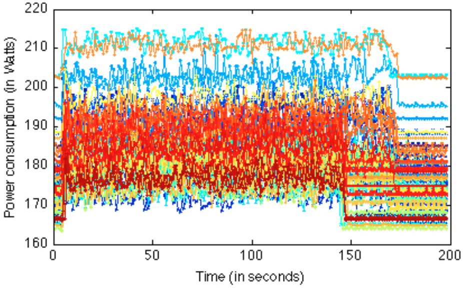


Figure: Power consumption of 10 GBytes of HDD logging

Energy in fault tolerance protocols - HDD logging

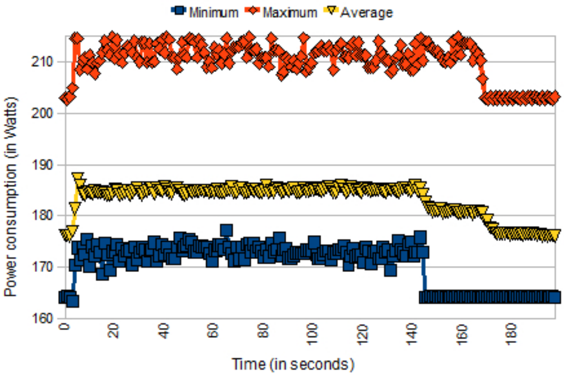


Figure: Power consumption of 10 GBytes of HDD logging

Energy in fault tolerance protocols - Message logging

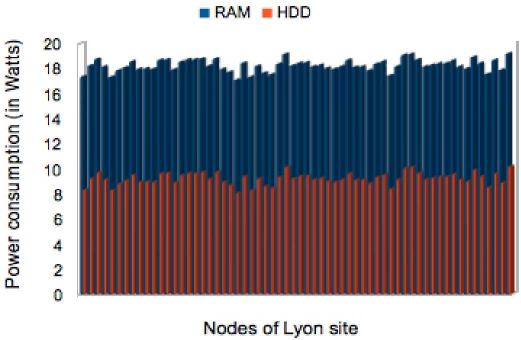


Figure: Extra power cost due to the message logging

RAM logging consume more power than HDD logging.

Energy in fault tolerance protocols - Message logging

Table: Energy ratio of Message Logging in Joules/GByte

Nodes	RAM logging energy consumption	HDD logging energy consumption
the less consuming	128	2550
the median consuming	137	2900
the more consuming	155	3600

We consume more energy by logging on HDD.
Values for message logging on HDD are close to those of checkpointing on disk.

Energy in fault tolerance protocols - Message logging

With a power capping point of view, users could decide to promote logging operation on HDD.

It is more energy efficient to log on RAM rather than on HDD. This is mainly due to the logging time:

- on HDD = more than 140 seconds for 10 GBytes
- on RAM = 7 seconds for 10 GBytes.

Energy in fault tolerance protocols - Process coordination

The process coordination implemented in MPICH2: a synchronization barrier.

a barrier in MPICH2 = an infinite loop that stops once the processes are synchronized.

In our testbed, an infinite synchronization barrier between the 64 processes of the 64 nodes

63 processes are running a barrier and 1 process is finalizing the MPI program.

We stop the infinite barrier after 30 seconds.

Energy in fault tolerance protocols - Process coordination

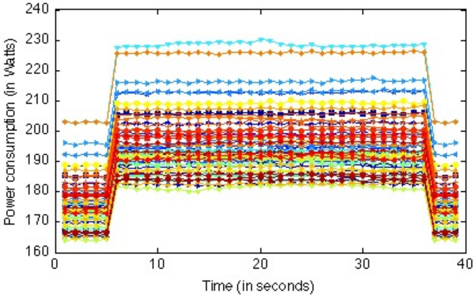


Figure: Power consumption of 64 nodes coordination

Energy in fault tolerance protocols - Process coordination

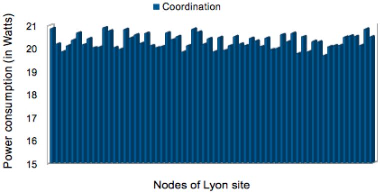


Figure: Extra power cost due to coordination for the 64 nodes

What is important is how long the coordination lasts.
eq. how long processes stay waiting each others.

We should minimize this waiting time:
slowing down the fastest processes (DVFS).

Results analysis - Comparison with intensive-using resources

Existing benchmarks that use resources (CPU, ...) intensively.

30 seconds of burnK6: an intermediate CPUburn [5].

30 seconds of HDparm [6].

30 seconds of STREAM [7]

[5] http://packages.debian.org/stable/cpuburn

[6] http://doc.ubuntu-fr.org/hdparm

[7] http://www.cs.virginia.edu/stream/

Results analysis - Comparison with intensive-using resources

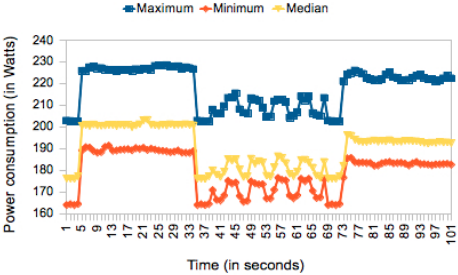


Figure: Power consumption for the most/less/median consuming nodes

Results analysis - Comparison with intensive-using resources

Table: Extra power cost comparison

HDparm between 7W and 10W	HDD Checkpointing and Logging 6W and 8W
STREAM 18W	RAM Logging 18W
burnK6 23W	Process coordination 20W

Energy-aware choice of fault tolerance protocols

Experiments with 4 NAS [8] in class C (BT, CG, LU, and SP).

RAM logging compared to Process coordinations.

Table: Overall extra energy consumption (in kJ) of RAM logging and coordinations in NAS benchmarks with 64 processes

	BT	CG	LU	SP
RAM logging	16.06	14.44	5.85	25.65
Coordinations	20.32	15.14	13.18	16.52

[8] http://www.nas.nasa.gov/publications/npb.html

Conclusion

Energy evaluation for fault tolerance protocols:

3 operations: checkpointing, message logging and coordination.

Process coordination and RAM logging consume more power than checkpointing and HDD logging.

For identical nodes performing the same operation, the extra power cost due to this operation is the same.

Power consumption of a node during a given operation remains constant during a operation.

Conclusion and Future Works

More power to store data on RAM.
HDD logging is more energy consuming than RAM logging.

We obtain the same extra power consumption for existing benchmarks that use intensively the same resources.

Proposed how to make an energy-aware choice of fault tolerance protocols:

Message logging should be preferred for applications involving small volumes of data exchanged.

Investigate energy efficient solutions for fault tolerance protocols by applying some green leverages.

Conclusion

Thank you for your attention.